# A Study on Relaying Soft Information with Error Prone Relays

Mehdi Mortazawi Molu
Institute of Telecommunications
Vienna University of Technology
Gusshausstr. 25/E389, 1040 Vienna, Austria
Email: mmortaza@nt.tuwien.ac.at

Norbert Goertz
Institute of Telecommunications
Vienna University of Technology
Gusshausstr. 25/E389, 1040 Vienna, Austria
Email: norbert.goertz@nt.tuwien.ac.at

***Abstract*—Cooperative communication is an effective way of achieving spatial diversity when it is impossible to employ multiple antennas. Recently, a new relaying technique, so called Soft Decode and Forward, has been proposed, which aims to combine the benefits of conventional relaying techniques, i.e. "Decode and Forward" and "Amplify and Forward", and to mitigate the shortcomings of those conventional methods. We study the error performance of two cooperative scenarios employing soft-DF. We propose a novel approach to estimate the bit-error probability and the equivalent channel SNR for relaying techniques of consideration in the paper that employ the BCJR-algorithm for decoding. We also consider the mutual information loss due to applying different encoders at the relay.**

## I. Introduction

Spatial transmit diversity by employing multiple antennas in the transmitter is one of the solutions to combat fading in wireless channels. In spite of the promising theoretical results, implementing multiple antennas in the user nodes can be practically infeasible, if not impossible, e.g. due to lack of space. A more recent approach to exploit spatial diversity is cooperation: several users work together to communicate with a common destination or even different destinations, so they can utilize transmit diversity by sharing resources and obtain better performance, i.e., higher throughput or lower error rates [1], [2], [3]. Two well-known relaying functions (e.g. [4]) are "Decode and Forward" (DF) and "Amplify and Forward" (AF). A more recent relaying function is "Soft Decode and Forward" (soft-DF), e.g. [5], [6], [7], where the intention is to combine the benefits of AF and DF and at the same time mitigate the shortcomings of the traditional algorithms. So far, the soft-DF technique has been evaluated in a scenario where distributed turbo coding [8] is applied. However, there is lack of literature evaluating recent soft-DF algorithm in simpler scenarios where the destination e.g. simply employs the Maximal Ratio Combining (MRC) technique instead of applying advanced iterative decoding algorithms.

There is plenty of literature explaining the Soft-Input Soft-Output BCJR encoder, e.g. [5], [6], [7]. Therefore we will not explain it again but rather assume that the reader is familiar with the concept.

The paper is organized as follows: in Section II we introduce our system model. In Section III we present methods to evaluate the performance of the system using hard and soft channel encoders at the relay. We present simulation results in Section IV and finally in Section V we try to explain the contradiction on the results of the two scenarios introduced in III.

## II. System Model

We consider a cooperative scenario in which a source node communicates with a destination via an intermediate relay node. In the sequel, we introduce two approaches of the cooperative scenario:

### A. *Case 1*

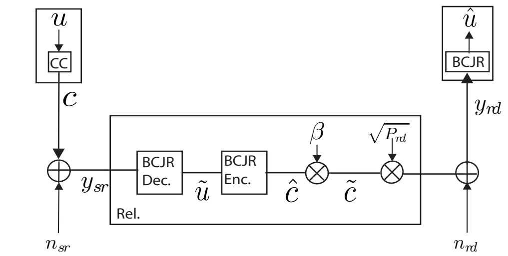
Fig. 1(a) shows the soft-relaying system under consideration. We assume that there is *no* direct link between the source and the destination, which can, e.g., be due to large distance between the source and the destination.

In the source node, a block of $K$ data bits is encoded using a rate-$k/n$ convolutional encoder, modulated and transmitted towards the relay. The relay employs a SISO BCJR decoder for decoding the noisy codeword received via the source-relay link. The output of the BCJR decoder is fed into a soft channel encoder, i.e. a SISO BCJR encoder. Its output is scaled by the factor $\beta$ to fulfil the power constraint of the relay and transmitted towards the destination. The destination employs the corresponding BCJR decoder for decoding the noisy codeword received via the relay-destination link.

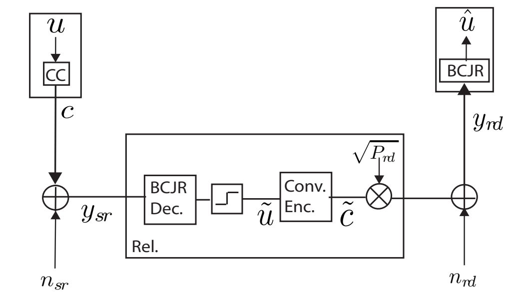
We deliberately make the assumption that the destination does not "hear" the source transmission in order to evaluate *only* the effect of the soft information produced by the relay on the performance, thereby studying the concept of soft channel encoding as such, without mixing the concept with (sub-optimal) iterative decoding in a distributed Turbo coding scheme (which will be further discussed in *Case 2*). This is also the reason why we use a simple convolutional code, as in this case an optimum symbol-by-symbol decoder (BCJR algorithm) is available [9].

### B. *Case 2*

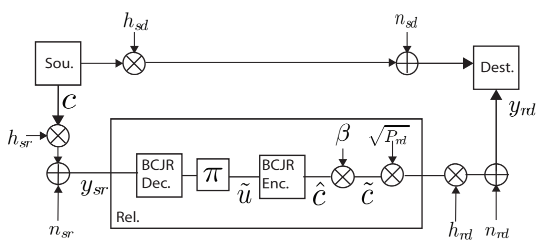
Extending *Case 1* to a more advanced scenario, we assume that there is also a direct link between the source and the




(a) Soft DF, employing a BCJR soft channel encoder




(b) Hard DF, employing a conventional channel encoder




(c) Distributed Turbo code employing a soft encoder

Fig. 1. System Model. For (a) and (c), $\beta$ is chosen such that the average power of the signal $\tilde{c}$ is normalised to one. In (b) the convolutional encoder includes BPSK modulation of the bits to $\pm 1$ signal values.

destination. The intention is to construct a Distributed Turbo Code (DTC) [8] applying soft information relaying [5], [6], [7]. Fig. 1(c) shows the corresponding soft-relaying system: the source operates as in the *Case 1*; the difference is that both the relay and the destination overhear the code word transmitted from the source. Since a Turbo code is applied for the overall system, we employ a RSC[1] encoder in the source. The relay decodes the received noisy codeword as in *Case 1* and interleaves the LLR[2]-values prior to soft encoding. The relay then employs a RSC Soft-Input Soft-Output (SISO) BCJR encoder for encoding permuted data symbols coming out of the BCJR decoder. The systematic bits of the relay encoder are punctured and only the parity check bits are sent to the destination. Prior to transmission, power related constraints, as explained in *Case 1*, are applied to the parity check symbols.

We assume AWGN[3] in all the source-relay, source-destination and relay-destination links. For simplicity we employ BPSK modulation in the source and the relay, but the magnitude of the transmitted BPSK modulation symbols is weighted according to the power scaling by the soft-encoded magnitudes of the code bits.

The signals received directly from the source at the relay and the destination at each BPSK symbol time instant are given by

$$y_{\rm sr} = \sqrt{P_{\rm sr}} \cdot h_{\rm sr} \cdot c + n_{\rm sr}, \quad c \in \{\pm 1\} \tag{1}$$

and

$$y_{\rm sd} = \sqrt{P_{\rm sd}} \cdot h_{\rm sd} \cdot c + n_{\rm sd}, \quad c \in \{\pm 1\} \tag{2}$$

respectively, and the signal from the relay received at the destination equals

$$y_{\rm rd} = \sqrt{P_{\rm rd}} \cdot \beta \cdot h_{\rm rd} \cdot \hat{c} + n_{\rm rd}, \tag{3}$$

where $\beta = 1/\sqrt{\overline{|\hat{c}|^2}}$, with $\overline{|\hat{c}|^2}$ the average power of the transmitted channel symbols, averaged over each block of soft-encoded code bits resulting from each block of $K$ data bits. Note that in *Case 2* the length of the relay codeword is half that of *Case 1*, as only the parity check bits are transmitted during the relay transmission. For simplicity, in *Case 1*, we assume a non-fading scenario, so $h_{\rm sr} = 1$ and $h_{\rm rd} = 1$. The noise components, $n_{\rm sr}$ and $n_{\rm rd}$, are zero mean real Gaussian random variables with variance $N_0$. For *Case 2* we assume Rayleigh fading where $h_{\rm sr}$, $h_{\rm sd}$ and $h_{\rm rd}$ are zero mean complex Gaussian random variable with variance $\sigma_{\rm h}^2$. The noise components, $n_{\rm sr}$ and $n_{\rm rd}$, are zero mean complex Gaussian random variables with variance $N_0$. The reason for the assumption of a Rayleigh fading channel for the case of distributed turbo codes will be explained in Section V.

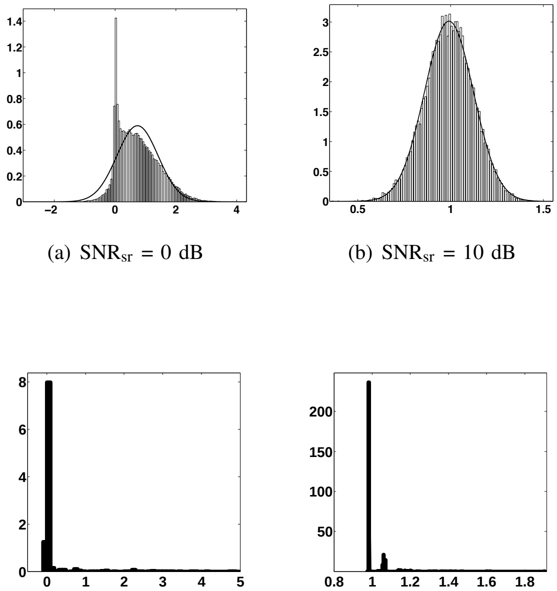
In Fig. 1(b) we show the competing system design for the scenario of *Case 1* using hard decisions for the data bits after soft-input channel decoding at the relay, prior to hard re-encoding (by a classical convolutional encoder), modulation and transmission to the destination. A similar competing hard-relay model is available for *Case 2*, although we omit that figure due space limits.

The receiver in *Case 1* employs a conventional BCJR decoder corresponding to the encoder of the relay. The receiver in *Case 2* applies iterative turbo decoder to decode the codeword received partially from the source-destination link and partially from the relay-destination link (Fig. 7).

## III. Performance Evaluation of SISO BCJR Encoder

One of the problems in cooperative communications is how to combine at the destination the available data coming from the source and the relay. The main intention of this section is to evaluate the performance of the SISO BCJR encoder. The difference of the two scenarios, *Case 1/2* (see Section II) is that the *Case 2* scenario applies an iterative Turbo decoder at the destination. This will influence the overall error performance due to the sub-optimality of the iterative decoder, regardless whether hard or soft encoding has been applied at the relay. Although Turbo codes are appreciated for their exciting error

[1] Recursive Systematic Convolutional code
[2] Log-Likelihood Ratio
[3] Additive White Gaussian Noise



(a) $\mathrm{SNR}_{\mathrm{sr}}$ = 0 dB (b) $\mathrm{SNR}_{\mathrm{sr}}$ = 10 dB

(c) $\mathrm{SNR}_{\mathrm{sr}}$ = 0 dB (d) $\mathrm{SNR}_{\mathrm{sr}}$ = 10 dB

Fig. 2. First row:Conditional probability density functions $p(\tilde{c}|c=+1)$ for a (7,5) FF SISO BCJR encoder. Second row:Conditional probability density functions $p(\tilde{c}|c=+1)$ for a (5/7) RSC SISO BCJR encoder. Power normalized to 1.

correction capabilities in a point-to-point scenario, in the case of DTCs, where two different nodes – the source and the relay – construct the turbo code, error prone relays can destroy the overall performance whenever the relay forwards erroneous data (code word fragments) to the destination. The problem gets even worse because of error propagation with every decoding iteration. This effect will be discussed in more detail in Section V.

In the *Case 1* scenario a simple BCJR decoder is used at the destination for decoding the received noisy codeword transmitted from the relay, without taking into account the source-destination link. Therefore, we evaluate the performance of SISO BCJR encoder assuming the *Case 1* scenario. We are interested in two parameters:

1) The mutual information loss due to soft/hard encoding at the relay.
2) The received SNR at destination due to relay transmission.

For proper decoding of the received noisy channel codeword, the BCJR decoder at the destination needs to know the statistics of the received signal that will depend on the channel as well as on the (soft) relaying function used. As illustrated by Fig. 1, $y_{\mathrm{rd}} = \tilde{c} + n_{\mathrm{rd}}$, where $n_{\mathrm{rd}}$ is a zero mean Gaussian random variable with variance $N_0$ (receiver noise). To the best of our knowledge, there is no closed form solution for the probability density function (pdf) of $\tilde{c}$ for non-trivial relaying functions. Hence, we have measured histograms that describe the conditional pdfs $p(\tilde{c}|c=1)$; the pdfs $p(\tilde{c}|c=-1)$ would be symmetric.

As an example, Figs. 2(a) and 2(b) show the pdfs $p(\tilde{c}|c=1)$ for different source-relay channel SNRs when a feed-forward BCJR soft channel encoder is applied in the relay, whereas Fig. 2(c) and 2(d) show the pdfs $p(\tilde{c}|c=1)$ when an RSC BCJR soft channel encoder is applied in the relay.

There is no analytical description relating the pdf of the output codeword $\tilde{c}$ to input pdf $p(\tilde{u})$. In literature (e.g. [5], [7]) the pdfs are usually modeled by zero mean Gaussian random variables $n_{\tilde{c}}$ to which a non-zero mean $\mu_{\tilde{c}}$ is added, with a sign that depends on the input bit $c \in \{-1,+1\}$, i.e.,

$$\tilde{c} = \mu_{\tilde{c}} c + n_{\tilde{c}}, \quad n_{\tilde{c}} \sim \mathcal{N}(0, \sigma_{\tilde{c}}^2)\,. \tag{4}$$

As obvious from the figures, the Gaussian assumption is not accurate especially for RSC BCJR encoder. In the remainder of this section we assume that a FF BCJR encoder is employed in the relay. Nevertheless, we will apply a RSC BCJR encoder when considering the *Case 2* scenario in forthcoming sections. As illustrated by Fig. 2(a), the Gaussian assumption also is not accurate at low $\mathrm{SNR}_{\mathrm{sr}}$ and, therefore, a performance degradation for low SNR is expected. To evaluate the performance of the soft channel encoding algorithms, we start by analysing the mutual information loss that occurs, due to the use of a soft channel encoder in the relay. We continue with the evaluation of the estimated BER of the overall system, and, based on that, we calculate the equivalent "receive SNR" at the destination. We compare the results of "soft" and "hard" relaying (algorithms as shown in Fig. 1).

### A. Mutual Information (Loss)

Mutual information, $I(U;\tilde{U})$, can be used to measure the amount of the information that soft (or hard) data bits, $\tilde{u}$, at the relay carry about the data symbols, $u$, transmitted by the source. As illustrated by Fig. 1 (a and b), the two system models, *soft/hard* DF, use two different (*soft/hard*) channel encoders. The intention of calculating mutual information is to measure the mutual information loss, [10], due to different channel encoders.

The mutual information $I(U;\tilde{U})$ [10], [11] between the (binary) transmitted data bits, $u \in \{+1,-1\}$, and the L-values $\tilde{u}$ is given by

$$\begin{aligned} I(U;\tilde{U}) = \frac{1}{2} \cdot \sum_{u'=\pm 1} \int_{-\infty}^{+\infty} p(\tilde{u} \mid u = u') \\ \times \mathrm{ld} \frac{2 \cdot p(\tilde{u} \mid u = u')}{p(\tilde{u} \mid u = +1) + p(\tilde{u} \mid u = -1)} d\tilde{u}\,, \end{aligned} \tag{5}$$

with $p(\tilde{u} \mid u)$ the conditional pdf of the L-values at the relay (see Fig. 1) given the input bits $u$. We have measured this pdf, similarly as the ones for the code bits ($p(\tilde{c}|c=1)$) but we have omitted the plots due to lack of space.

To characterize $I(U;\tilde{U})$ associated with hard-DF, we model the source-channel-relay link as a Binary Symmetric Channel (BSC) in which the channel input is a data bit. The output bit of the channel is flipped with probability $q$. Hence, mutual information for such a BSC is given (e.g. [10]) by

$$I(U;\tilde{U}) = 1 - H_2(q), \qquad (6)$$

with $H_2(q) \doteq -q \cdot \mathrm{ld}(q) - (1-q) \cdot \mathrm{ld}(1-q)$ the standard binary entropy function.

Using the same approach, one can calculate the mutual information $I(C;\tilde{C})$, too. A comparison of the mutual informations $I(U;\tilde{U})$ and $I(C;\tilde{C})$ for both the *hard/soft* DF is useful in characterizing the performance of the soft channel encoders. Numerical results will follow in Section IV.

### B. Equivalent Receive SNR at the Destination

The bit error rates at the destination and the equivalent receive SNR (equivalent for a substitute AWGN channel) at the destination are closely related, with the equivalent receive SNR providing extra insight into the communication process.

*1) Hard DF:* Calculating the equivalent receive SNR at the destination for hard DF is somewhat cumbersome, due to the error-prone relay. Since the relay decodes and forwards both the correct and erroneous frames, the distribution of the received signal at destination is no longer Gaussian. The common approach to estimate the SNR at the destination is to model the source-relay-destination link as an equivalent AWGN channel with channel $\mathrm{SNR}_{\mathrm{eq}}$ that depends on both the source-relay and the relay-destination channel qualities.

The total bit error probability is given by

$$\begin{aligned} P_{\mathrm{tot}}(e \mid \gamma_{\mathrm{sr}}, \gamma_{\mathrm{rd}}) &= P_b(e \mid \gamma_{\mathrm{sr}})[1 - P_b(e \mid \gamma_{\mathrm{rd}})] \\ &\quad + [1 - P_b(e \mid \gamma_{\mathrm{sr}})] P_b(e \mid \gamma_{\mathrm{rd}}), \end{aligned} \qquad (7)$$

where $\gamma$ and $P_b(e)$ are the corresponding channel SNR and the bit error probabilities for the two links (source-relay and relay-destination) involved.

Calculating $P_{\mathrm{tot}}$ using simulations is straightforward but one can also calculate it using the complementary error function. The bit error probability of convolutional codes under symbol-by-symbol MAP decoding can be approximated by

$$P_b(e) \approx \frac{1}{2}\mathrm{erfc}\Big(\sqrt{\frac{\mu_{\mathrm{out}}^2}{2\sigma_{\mathrm{out}}^2}}\Big) = \frac{1}{2}\mathrm{erfc}\Big(\sqrt{\frac{1}{2}\gamma_{\mathrm{out}}}\Big), \qquad (8)$$

(e.g. [11]) where $\mu_{\mathrm{out}}^2$ and $\sigma_{\mathrm{out}}^2$ are, respectively, the mean and variance of the data bit L-values at the output of the BCJR decoder, and $\gamma_{\mathrm{out}} = \mu_{\mathrm{out}}^2/\sigma_{\mathrm{out}}^2$. Similarly, $\mu_{\mathrm{in}}^2$ and $\sigma_{\mathrm{in}}^2$ would define $\gamma_{\mathrm{in}} = \mu_{\mathrm{in}}^2/\sigma_{\mathrm{in}}^2$ for the input L-values of the symbol-by-symbol MAP decoder.

Fig. 3 illustrates $\gamma_{\mathrm{out}} = f(\gamma_{\mathrm{in}})$ for a (7,5) convolutional code, decoded by a BCJR decoder. Using regression analysis, $\gamma_{\mathrm{out}}$ can be modeled by a polynomial according to

$$\begin{aligned} \gamma_{\mathrm{out}} &= f(\gamma_{\mathrm{in}}) \\ &\approx 3.38 \cdot 10^{-3}\gamma_{\mathrm{in}}^3 - 0.12\gamma_{\mathrm{in}}^2 + 2.38\gamma_{\mathrm{in}} + 2.56. \end{aligned} \qquad (9)$$

$P_b(e \mid \gamma_{\mathrm{sr}})$ and $P(e \mid \gamma_{\mathrm{rd}})$ in (7) can be computed using (8) and (9) (with $\gamma_{\mathrm{in}} \in \{\gamma_{\mathrm{sr}}, \gamma_{\mathrm{rd}}\}$). Given $P_{\mathrm{tot}}$, the equivalent $\mathrm{SNR}_{\mathrm{out}}$, $\gamma_{\text{eq-out}}$, follows from

$$\gamma_{\text{eq-out}} = 2\Big(\mathrm{erfc}^{-1}(2P_{\mathrm{tot}})\Big)^2. \qquad (10)$$

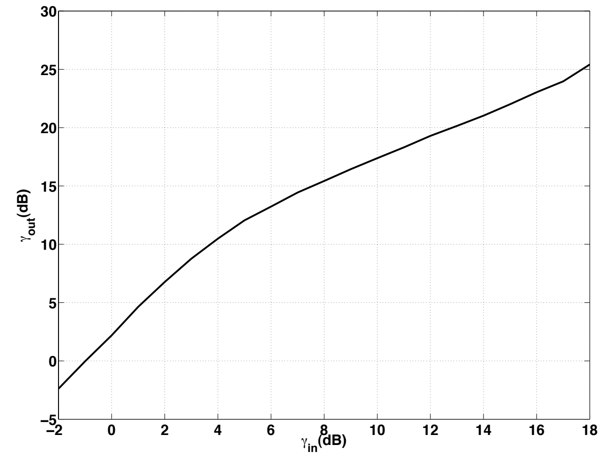



Fig. 3. $\gamma_{\mathrm{out}} = f(\gamma_{\mathrm{in}})$ for a (7,5) convolutional code under symbol-by-symbol MAP (BCJR) decoding

By substituting (10) into

$$\gamma_{\mathrm{eq}} = f^{-1}(\gamma_{\text{eq-out}}), \qquad (11)$$

the equivalent source-relay-destination SNR, $\gamma_{\mathrm{eq}}$, is computed.

*2) Soft DF:* One might exploit the Gaussian assumption of (4) for calculating $\mathrm{SNR}_{\mathrm{eq}}$ of the soft DF schemes, but, as illustrated by Fig. 2, the Gaussian assumption is not accurate, especially at low SNR. Therefore, in order to calculate $\mathrm{SNR}_{\mathrm{eq}}$, we use Monte-Carlo simulations in the destination to determine $\gamma_{\text{eq-out}}$. With $\gamma_{\text{eq-out}}$, calculating $\gamma_{\mathrm{eq}}$ using (11) is straightforward. The estimated BER for soft DF can then be determined by substituting $\gamma_{\text{eq-out}}$ in (8).

## IV. Simulation Results

We start the simulations for the *Case 1* scenario. We will then present the results of the *Case 2* scenario and continue by explaining the reasons why the two scenarios show different error performances in spite of employing similar encoders at the relay.

### A. Case 1

The *Case 1* scenario, which is illustrated by Fig. 1(a), has the following characteristics: the source applies a [7,5] convolutional encoder for encoding information frames of length 2000 bits; for simplicity we also use BPSK modulation. The modulated code symbols are scaled by the source's transmit power constraint and are sent to the relay. We assume that receive signal in the relay is corrupted by zero mean real Gaussian receiver noise with a variance of $N_0 = 1$. In order to decode the received noisy codeword, the relay employs a standard soft-in/soft-out symbol-by-symbol MAP (BCJR) decoder [9]. The L-values of the BCJR-decoded information bits (or equivalently their a-posteriori probabilities) are then used for calculating the L-values of the code bits by the BCJR soft channel encoder. For simplicity the BCJR encoder has the trellis structure of the [7,5]-convolutional code, although other

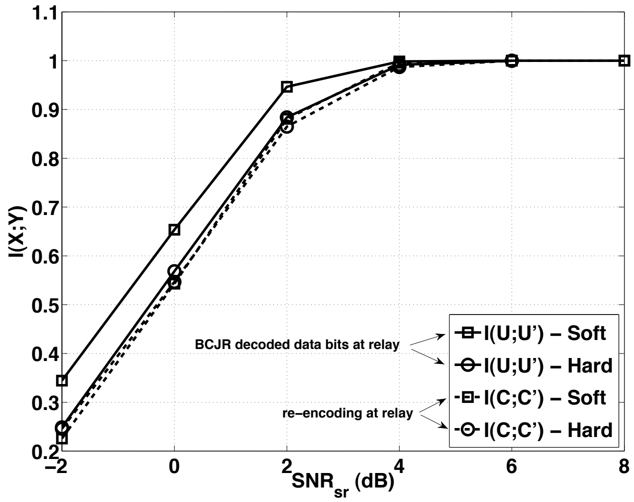



Fig. 4. Mutual information for hard and soft channel encoding compared with mutual information between the original data bits $U$ and the BCJR soft-decoded data bits $U'$ (with soft and with hard decisions after decoding) at the relay prior to re-encoding.

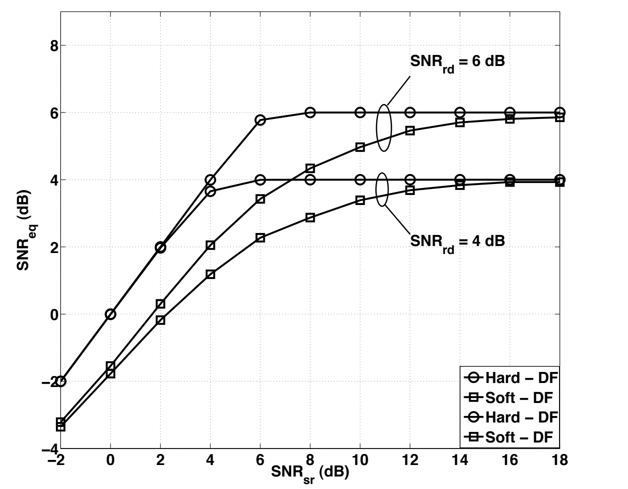



Fig. 5. Equivalent source-relay-destination SNR; for soft/hard DF, a (7,5) SISO BCJR channel encoder is used.

codes, and hence other trellis structures, could be used. The output L-values for the code bits of each codeword are then normalized by $\beta$ in order to fulfil a unit-power constraint, so the same average power as in conventional "hard" BPSK modulation is used. The unit-power code-bit L-values are then scaled by the (square-root) of the relay's transmit power constraint ($P_{\mathrm{rd}}$) and transmitted towards the destination. The destination applies a BCJR decoder and takes hard decisions to obtain the output data bits $\hat{u}$.

We also simulate a scenario, in which the relay performs conventional DF, i.e., hard encoding at the relay (Fig. 1(b)).

Fig. 4 compares the mutual informations $I(U;\tilde{U})$ of the data bits $U$ and their decoded counterparts $\tilde{U}$ (with soft and hard decisions after soft-input BCJR decoding, see Figs. 1(a) and 1(b) at the relay as well as the mutual informations $I(C;\tilde{C})$ of the code bits $C$ at the source and the (soft and hard) re-encoded code bits $\tilde{C}$ at the relay. As expected, the mutual information $I(U;\tilde{U})$ for the soft-DF scheme is larger than for hard-DF. This confirms the received wisdom that "soft is better than hard". The reason is the quantization (for hard DF) applied on the L-values of the data bits at the output of the BCJR decoder in the relay: this quantization obviously destroys information.

In hard-DF the relay applies a conventional convolutional encoder to compute the transmitted code bits but in soft-DF a (BCJR) soft channel encoder is used. Because of applying different encoders, the mutual information loss[4] explained in Section III-A will be different. As illustrated by Fig. 4, the mutual information loss is more severe for soft-DF than for hard DF. In fact, although the input data to the channel encoder of soft-DF contains more information than the input data of hard-DF, there is still a slight mutual information loss at the output of the convolutional encoder. This indicates that the soft channel encoding scheme actually destroys more information by data processing than the hard encoding algorithm which has less information at its input. Fig. 5 shows the equivalent channel SNR for both the hard DF algorithm and the soft DF algorithm using a BCJR soft channel encoder. The $\mathrm{SNR}_{\mathrm{eq}}$ curve of soft DF merges with the $\mathrm{SNR}_{\mathrm{eq}}$ curve of hard-DF at high $\mathrm{SNR}_{\mathrm{sr}}$, although a small difference remains. The explanation is that the relay usually performs error free decoding at high SNR; therefore the hard-DF algorithm is very close to optimum at high SNR. But for soft-DF, the transmitted symbol from the relay, $\tilde{c}$, is Gaussian distributed. Because of the unit power constraint of BPSK modulation that we have to enforce for soft encoding in an average sense as well, we have an average power of $\mathrm{P}(\tilde{c}) = \mu_{\tilde{c}}^2 + \sigma_{\tilde{c}}^2 = 1$. Since $\sigma_{\tilde{c}}^2 > 0$, we find that $|\mu_{\tilde{c}}| < 1$ must hold, so some of the transmitted code bits will have smaller instantaneous power than the hard-encoded symbols (which all have "one"): this will cause the slight $\mathrm{SNR}_{\mathrm{eq}}$ degradation in comparison with hard-DF for large values of $\mathrm{SNR}_{\mathrm{sr}}$. The $\mathrm{SNR}_{\mathrm{eq}}$ of the system is bounded by $\min\{\gamma_{\mathrm{sr}}, \gamma_{\mathrm{rd}}\}$, i.e. $\gamma_{\mathrm{eq}} < \min\{\gamma_{\mathrm{sr}}, \gamma_{\mathrm{rd}}\}$ .

Fig. 6 illustrates the bit error rates of the system for $\mathrm{SNR}_{\mathrm{rd}}$=4dB. The estimated curves for both the hard and soft DF (for BCJR soft channel encoding) confirm the simulation results. The largest BER gap between hard DF and soft DF appears in an $\mathrm{SNR}_{\mathrm{sr}}$ range between 3–7 dB, which corresponds to a similar gap shown in Fig. 5.

For the *Case 1* scenario similar results are obtained if a RSC SISO BCJR encoder is applied at the relay but due to space limits we omit the results. The reason for us to consider the FF encoder but not the RSC encoder is the Gaussian-like pdf of the FF SISO BCJR encoder's code symbols, which makes analysis much easier.

[4]Any processing of data can only cause a loss of information: data procession theorem, [10].

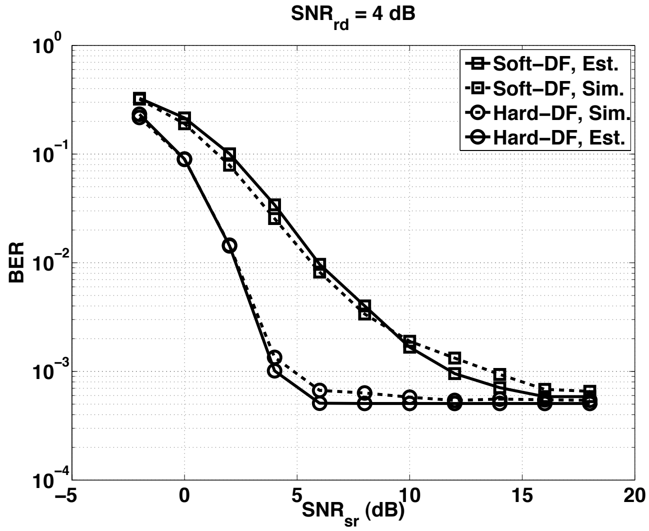



Fig. 6. Bit-error-rate performance of hard DF and soft DF for the *Case 1* scenario. Because of the limited quality of $\mathrm{SNR}_{\mathrm{rd}}$ = 4 dB of the relay-destination link, the curves saturate at large source-relay channel quality $\mathrm{SNR}_{\mathrm{sr}}$.

*B. Case 2*

The *Case 2* scenario, which is illustrated by Fig. 1(c), has the following characteristics: the frame length, modulation and noise characteristics are the same as in the *Case 1* scenario. Since the intention is to implement a distributed turbo coding scheme, we apply a [1, 5/7] RSC convolutional encoder in the source. We assume that all the channels $h_{sd}$, $h_{sr}$ and $h_{rd}$ are Rayleigh fading channels with unit variance. The relay employs the corresponding the BCJR decoder for decoding the noisy codeword received via the source-relay link without taking hard decisions. The relay permutes the LLR values of decoded information bits using a random interleaver. The permuted soft information is then used as a soft input for a [1,5/7] SISO BCJR encoder. The systematic bits are punctured and only the "soft-encoded" parity check bits are transmitted towards the destination. Of course, power normalization, as in the *Case 1* is applied before transmission. The receiver employs an iterative Turbo decoder for decoding the received codeword. Fig. 7 illustrates the turbo decoder used for our simulations. Every time data is passed through one of the BCJR components is called one iteration and we perform a fixed number of five iterations for every frame.

Fig. 8 shows the BER performance for the *Case 2* scenario when hard/soft encoding has been applied in the relay and $P_{sr} = P_{rd} =$ 12dB. The figure clearly shows that after 5 times iteration, soft relaying outperforms hard relaying with a considerable difference of about 10 dB. Such a performance behaviour has been also reported in other publications (e.g. [5], [6], [7]).

The interesting point is that for hard-DTC, iterations cause loss of performance so that performing iterations degrades performance compared to the case when there are no iterations and the destination operates only once on the source-destination channel's output (i.e. only BCJR 1 runs once). Apparently, the error performance of the *Case 2* scenario contradicts the *Case 1* scenario. In the *Case 1* scenario the "hard" algorithm outperforms the "soft" algorithm while in the *Case 2* scenario, the soft algorithm outperforms the hard algorithm. In the rest of the paper we will try to explain the reason for this.

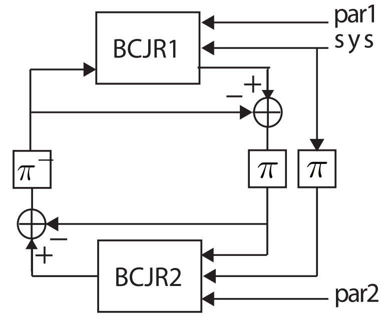



Fig. 7. Turbo decoder of *Case 2*. The data processed by BCJR 1 experiences the fading coeffcient $h_{sd}$. The systematic bits processed by BCJR 2 experience $h_{sd}$, too, but the parity check bits processed by BCJR 2 experience a different fading coefficient $h_{rd}$.

## V. Discussion

The analysis of DTC with error prone *soft/hard* relays is hard to handle when the relay fails to decode correctly. In order to simplify the problem we assume, without loss of generality, that the "all-zero" codeword is used. We start with conventional convolutional encoding at the relay, i.e., the relay decodes the noisy all zero codeword, interleaves the decoded bits and then encodes them using an RSC encoder and sends the parity check bits towards the destination. Assume the case that the relay fails decoding: since we use an RSC encoder at the relay, even one data bit in error will propagate along the entire frame which will cause a burst-error at the destination. Depending on the position of the data-bit error, the number of erroneous (re-encoded) parity check bits (with "1"-values) of the RSC encoder will differ. If the interleaved error bit occurs at the beginning of the frame, the entire frame will be corrupted by the propagating errorenous bit whereas, if the interleaved bit error occurs at the end of the frame, only a few codebits at the end will be "incorrectly encoded".

Fig. 9(b) illustrates the distribution of received symbols (parity check symbols) transmitted from the relay for a given AWGN channel with unit variance. The figure shows that *almost* half of the parity check bits transmitted from the relay are erroneous. When the relay fails decoding, the parity check bits at the output of the relay do not correspond to the original codeword. In fact, the combination of systematic bits transmitted from the source and parity check bits transmitted from the relay are unlikely to form a valid codeword at all. The systematic bits and parity bits are then combined in the destination to be used by decoder BCJR 2. However, there is no guarantee that the combination of these two sets of information will construct a valid codeword. Therefore, a theoretical

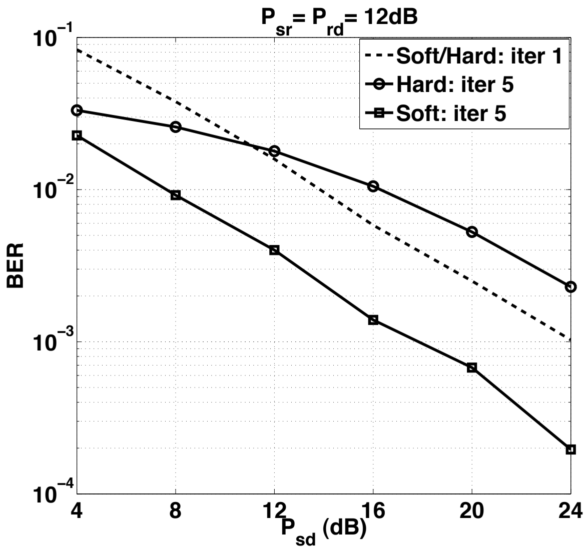



Fig. 8. BER for the *Case 2* scenario

analysis of distributed turbo codes using conventional methods (such distance properties) is very difficult (e.g.[12], [13]). That is why we resort to the distribution of the LLR values for analysis.

We will first consider the *hard* DTC by considering the distribution of LLR values of information bits at the output of the two BCJR decoders after every iteration. Fig. 9(a) shows the distributions of the received noisy codeword LLRs at the destination as received from the source; the first BCJR decoder (BCJR 1 in Fig. 7) decodes the message given this information and outputs the LLR values of the information bits. It is assumed that there is no a priori information available in the first iteration[5] but a priori information will be available for the following iterations.

The second BCJR decoder (BCJR 2 in Fig. 7) uses three sets of information for decoding:

1) a priori information of the data bits, calculated by BCJR 1
2) received signal at the destination, transmitted from the source, (Fig. 9(a)), corresponding to the systematic bits of the codeword
3) received signal at the destination, transmitted from the relay, (Fig. 9(b)), corresponding to the parity-check bits of the *supposed* codeword.

It is unlikely that such a codeword (formed from the sets of information available) will exist; in fact, for the all zero dataword we are sure that it is *not* a codeword because, given all zero systematic bits, there is no codeword with hamming weight larger than zero. Therefore, we expect that decoding will fail in decoder BCJR 2 when the relay fails to decode correctly. Fig. 9(c) shows the distribution of LLR values of the all-zero databits after the first iteration. The BCJR 2 component tries to decode databits given Figs. 9(a), 9(b) and a priori information 9(c) but it fails to decode correctly because of the problem described above. Decoding failure is evident from Fig. 9(d): the LLR values of data bits reach values as low as $-100$, when their signs "should" all be positive (to be correct). The erroneous a priori information will propagate through every iteration.

[5]every time running each of BCJR decoders is called "one iteration"

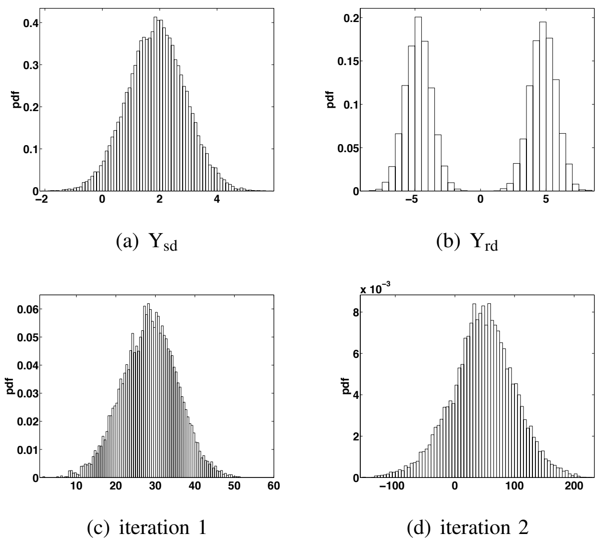
The situation is somewhat different for soft DTC. Fig. 2(c) illustrates that the reliability of soft encoded symbols in the relay tends to zero for low $\text{SNR}_{\text{sr}}$. Fig. 5 also shows larger received SNR at destination for hard encoding compared to SISO BCJR encoding. However, despite these facts, soft-DTC outperforms hard-DTC.

The reason why soft-DTC outperforms hard-DTC in spite of less reliable data transmitted from the relay lies behind the distribution of the signals received at the destination due to transmission from the relay. At low channel $\text{SNR}_{\text{sr}}$, the transmitted signal from the relay has very small reliability; see e.g. Fig. 2(c): at a first glance, it seems that this "information" is not worth to be transmitted, as the resulting signal received at the destination will be Gaussian distributed but with mean very close to zero (see Fig. 10(b)). In hard-DTC, the received signal at the destination (e.g. Fig. 9(b)) consists of two parts: the signals corresponding to correct bits (the right-hand set of signals in Fig. 9(b)) and those corresponding to the erroneous bits (the left-hand set of signals in Fig. 9(b)). The difference to soft-DTC is that in the hard-DTC the correct bits have large positive reliability (which is of interest) but the erroneous bits have large negative reliability.

The first iteration of the soft turbo decoder (Fig. 10(c)) wroks like the first iteration of hard turbo decoder (Fig. 9(c)). But in the second iteration of the soft turbo decoder, performance improves – albeit only slightly – because the parity check bits transmitted from the relay convey some – albeit very little – useful information. But in contrast to the hard DTC, erroneous information from the relay does not appear to the highly reliable, what is cause in hard DTC by hard encoding. Hence, with soft-DTC there is no such a striking performance degradation by further iterations. In other words: the second BCJR decoder of soft the Turbo decoder works based on three sets of data

1) a priori information of the data bits, which is calculated in BCJR 1
2) received signal at destination transmitted from the source (Fig. 10(a)) which is the systematic bits of the codeword.
3) received signal at destination transmitted from the relay (Fig. 10(b)).

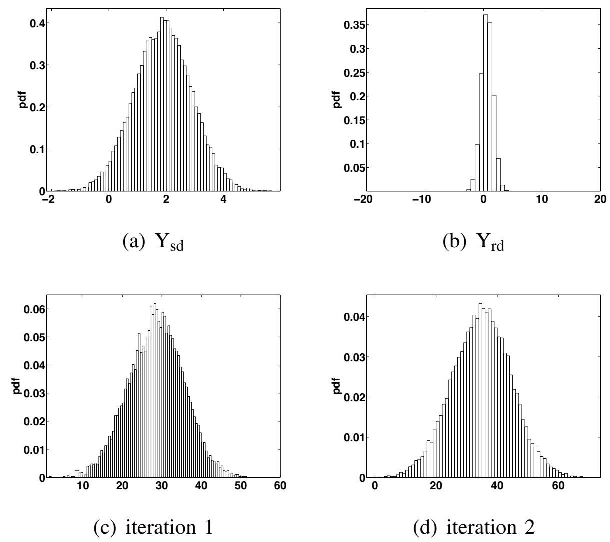
However, since the mean of the transmitted signal from the relay tends to zero (in the "error case"), this set of information will be treated as noise in the destination. Therefore even though the BCJR 2 does not considerably improve the error performance it also does not degrade the performance due to incorrect a priori information, unlike hard-DTC, in which seemingly reliable negative parity check symbols "confuse" the decoder BCJR 2. Fig. 10(d)clearly show that further iterations in the *Case 2* scenario, in spite of erroneously




(a) $Y_{sd}$ (b) $Y_{rd}$

(c) iteration 1 (d) iteration 2

Fig. 9. (a): distribution of the received signal at the destination with the "all-zero" codeword transmitted from the source.
(b): distribution of the received signal at the destination resulting from RCS convolutional encoding at the relay.
(c) and (d) distribution $p(L|c = +1)$ of the information-bit LLR values at the destination after the indicated number of iterations




(a) $Y_{sd}$ (b) $Y_{rd}$

(c) iteration 1 (d) iteration 2

Fig. 10. (a): distribution of the received signal at the destination with the "all-zero" codeword transmitted from the source.
(b): distribution of the received signal at the destination resulting from SISO BCJR encoding at the relay.
(c) and (d) distribution $p(L|c = +1)$ of the information-bit LLR values at the destination after the indicated number of iterations

decoding at the relay, cause a slight improvement in error performance.

## VI. Conclusions

We can divide the simulations into two parts: In a first case, the relay may successfully decode. Then hard-DTC outperforms soft-DTC.

In a second case, the relay may fail to decode correctly. Then, error bursts produced by hard-DTC dramatically destroy the performance. Using a Rayleigh fading assumption for the channel, we can be sure that with a certain (non-zero) probability there will be channel conditions in which the source-relay link operates at low SNR and, therefore, error bursts will indeed frequently destroy the performance of hard-DTC and it is exactly then when soft-DTC outperforms hard-DTC. Otherwise the SISO BCJR encoder does not outperform the convolutional encoder in the sense of mutual information loss nor SNR enhancement, as explained in Section III. The only advantage of the SISO BCJR encoder appears in the case of decoding failure at the relay. Then, it produces soft information which tends to zero and, hence, will not cause error propagation in a DTC scenario.

Overall, the better performance of soft-DTC is, therefore, not a property of the coding scheme as such but rather the consequence of the application of a sub-optimal, iterative and distributed decoding scheme.

## References

[1] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. part i. system description," *Communications, IEEE Transactions on*, vol. 51, no. 11, pp. 1927 – 1938, 2003.

[2] A. Sendonaris, E. Erkip, and B. Aazhang, "User cooperation diversity. part ii. implementation aspects and performance analysis," *Communications, IEEE Transactions on*, vol. 51, no. 11, pp. 1939 – 1948, 2003.

[3] J. Laneman, D. Tse, and G. Wornell, "Cooperative diversity in wireless networks: Efficient protocols and outage behavior," *Information Theory, IEEE Transactions on*, vol. 50, no. 12, pp. 3062 – 3080, 2004.

[4] K. S. Gomadam and S. A. Jafar, "Optimal relay functionality for SNR maximization in memoryless relay networks," *Selected Areas in Communications, IEEE Journal on*, vol. 25, no. 2, pp. 390 –401, 2007.

[5] H. Sneesens and L. Vandendorpe, "Soft decode and forward improves cooperative communications," in *Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE International Workshop on*, pp. 157 –160, 2005.

[6] Y. Li, B. Vucetic, T. Wong, and M. Dohler, "Distributed turbo coding with soft information relaying in multihop relay networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2040 –2050, 2006.

[7] P. Weitkemper, D. Wübben, V. Kühn, and K. Kammeyer, "Soft information relaying for wireless networks with error-prone source-relay link," in *Int. ITG Conf. on Source and Channel Coding*, 2008.

[8] M. Valenti and B. Zhao, "Distributed turbo codes: towards the capacity of the relay channel," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 1, pp. 322 – 326 Vol.1, 2003.

[9] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *Information Theory, IEEE Transactions on*, vol. 20, pp. 284 – 287, Mar. 1974.

[10] T. M. Cover and J. A. Thomas, *"Elements of Information Theory"*. A John Wiley & Sons, 2006.

[11] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *Communications, IEEE Transactions on*, vol. 49, pp. 1727 –1737, Oct. 2001.

[12] O. Takeshita, M. Fossorier, and D. Costello, "A new technique for computing the weight spectrum of turbo-codes," *Communications Letters, IEEE*, vol. 3, pp. 251 –253, aug 1999.

[13] S. Benedetto and G. Montorsi, "Unveiling turbo codes: some results on parallel concatenated coding schemes," *Information Theory, IEEE Transactions on*, vol. 42, pp. 409 –428, mar 1996.